CAE

SE CORE
SYNTHETIC ENVIRONMENT CORE

SE Core Database Virtual Environment Development (DVED)

Developers Forum

December 2006

ARINC YOU WON'T BELIEVE WHAT WE CAN DO.®

Rockwell Collins

LOCKHEED MARTIN
We never forget who we're working for™

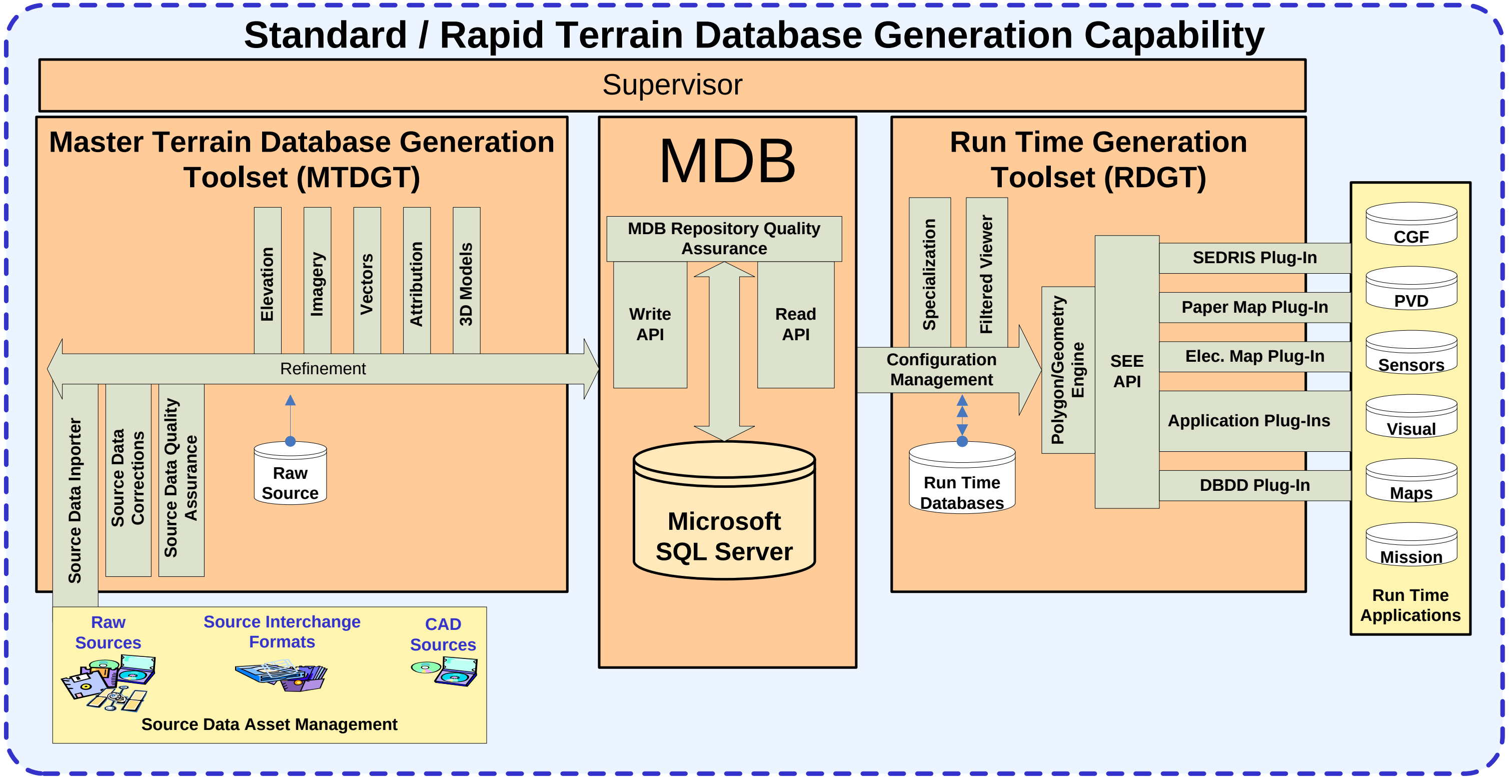
TERREX
TERRAIN EXPERTS, INC.

SECUREINFO

Topics

- DVED Architecture Overview
- Vendor & Program Plug-in Implementation Concept
- Case Study #1 – TOPScene Integration
- Case Study #2 – U2MG Integration
- Case Study #3 – Output Compiler Plug-in Development
- Developer Documentation & Support Approach
- Developer Support Milestones
- Questions?

DVED Architecture Overview



Plug-in Type: Tool Integration

- Example: TOPSCENE .image format import
  - Uses the Master DataBase (MDB) Read/Write API
  - Also can use the Common Environment Representation (CER) API, and import to the MDB through TerraVista.
  - Allows custom data formats to be imported and exported to the MDB
  - Once data is in the MDB, all DVED process tools can utilize it

Plug-in Type: Culture Compiler

- Example: You have a process that adds value to the synthetic environment, such as feature extraction or scattering of urban data
- These plug-ins are invoked during the database build process

Plug-in Type: Training Device Client

- Example: You are an IG vendor and you want DVED to be able to output your format natively
- The Synthetic Environment Exchange (SEE) API Allows vendors to add support for their runtime database support.
- Each output compiler in a build gets the same geometry and other data to produce the output.

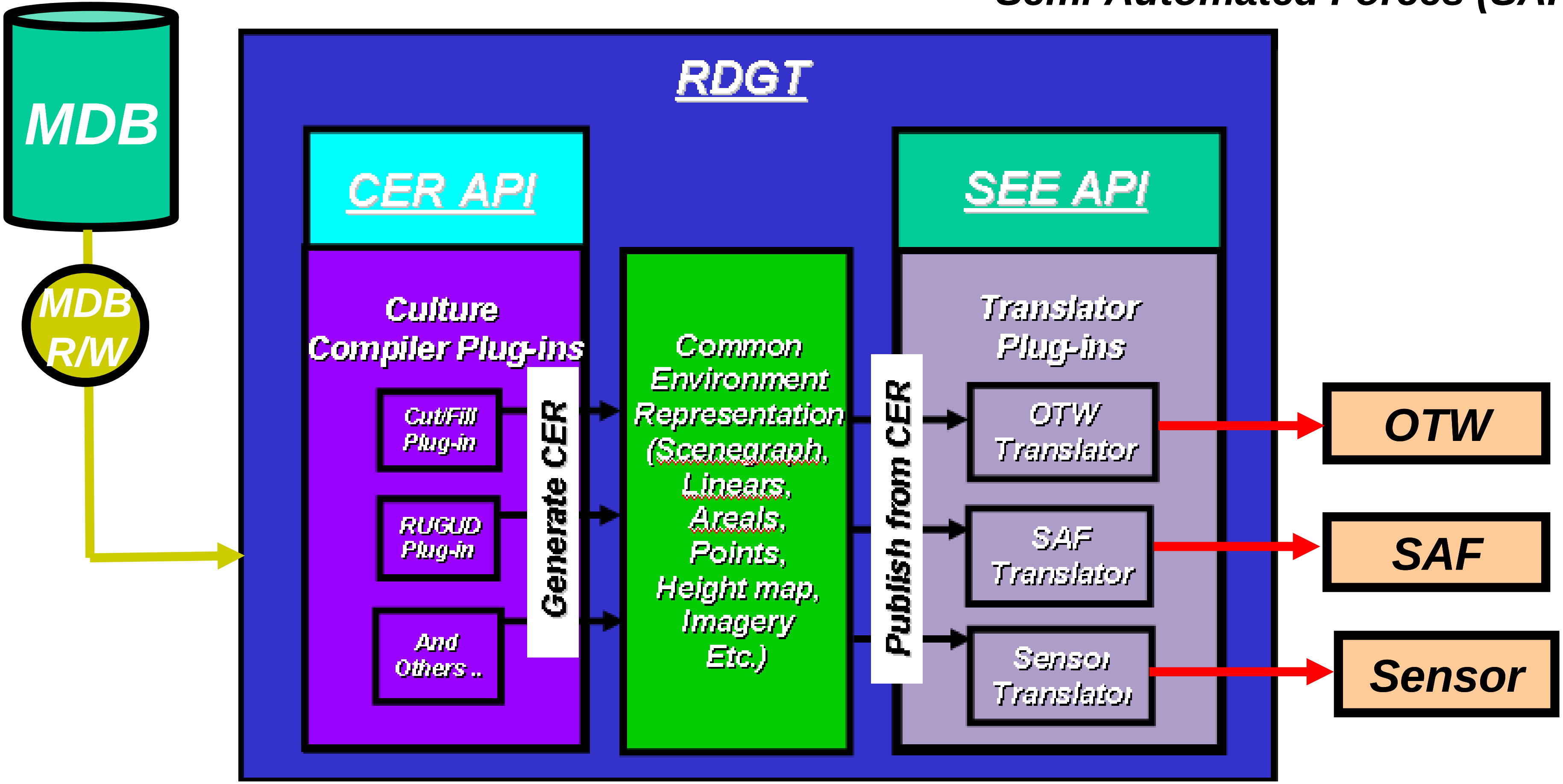
Vendor & Program Plug-in Implementation Concept

– Plug-in types

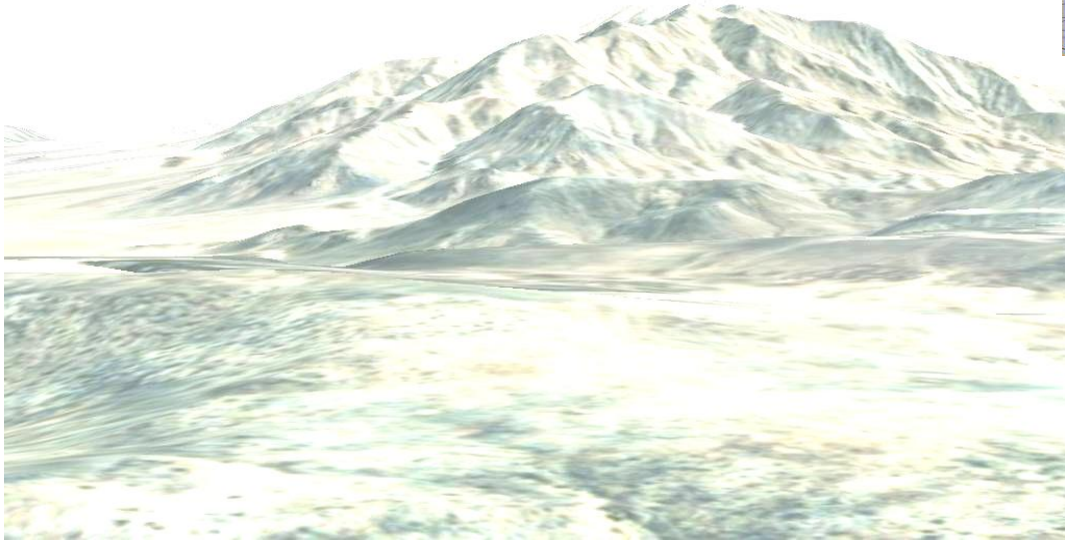
- Translator plug-ins (CER API) / (MDB Read/Write API)
  - **Allow vendors to add support for file formats to be ingested into the DVED tools**
  - **Tool vendors can read and write MDB data with their tools by using the MDB Read/Write API**
- Culture Compilers (CER API)
  - **Allow vendors to interact with the scene generation phase of the build process to add value (e.g. feature extraction, adding urban clutter, etc.)**
- Output Compilers (SEE API)
  - **Allows vendors to add support for their runtime database support**
  - **Each output compiler in a build gets the same geometry and other data to produce the output**

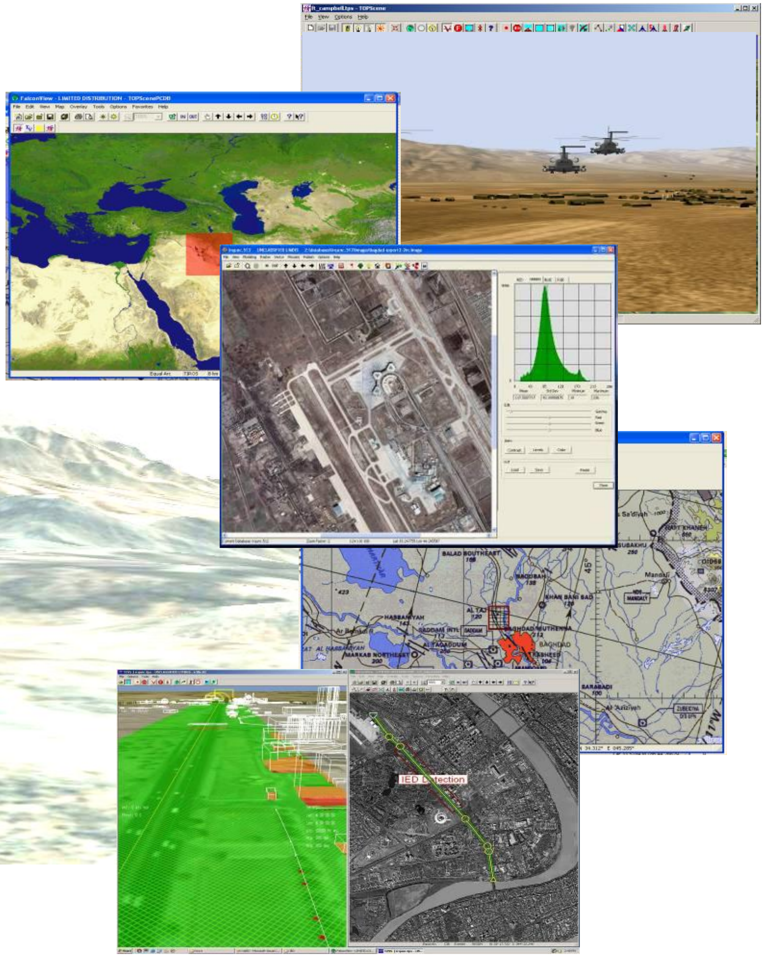
DVED Plug-in Types

Out the Window (OTW)
Semi Automated Forces (SAF)



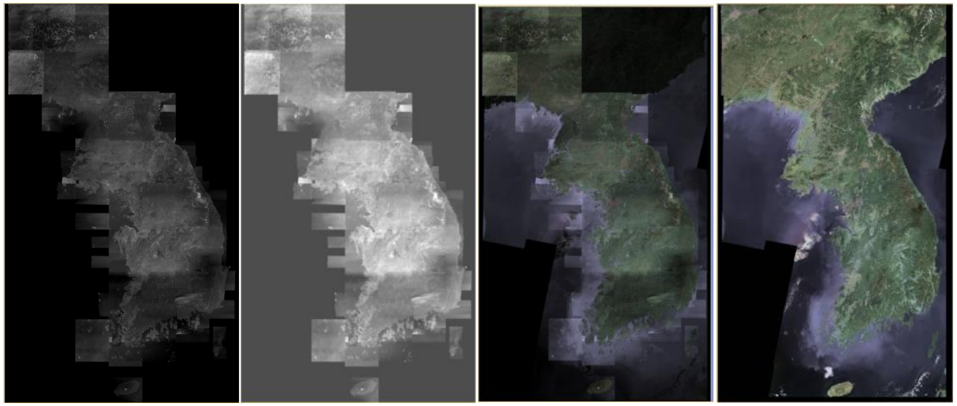
Case Study #1
TOPSCENE® Integration





Integration Overview

- TOPSCENE® Integrated with Master Database (MDB)
- Importers & Exporters Developed with MDB API
  - – Imagery, 3D Models, Vector Data, Raster Elevation
- Mission Rehearsal Support
  - – Rapid High Resolution Image Registration
  - – Rapid Geospecific 3D Target Modeling
- MDB Adopted as TOPSCENE® Archival Format



TOPSCENE MDB Integration

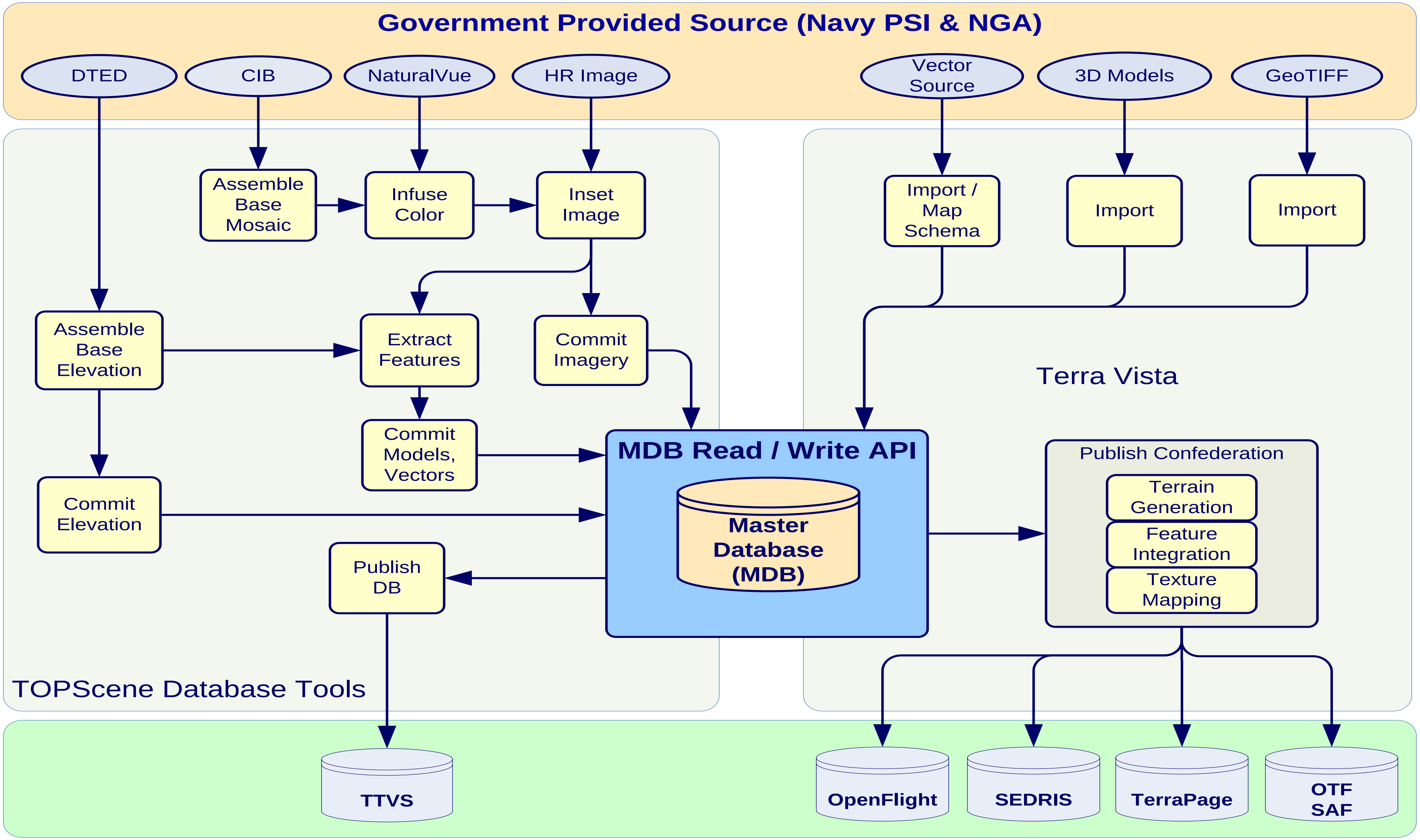
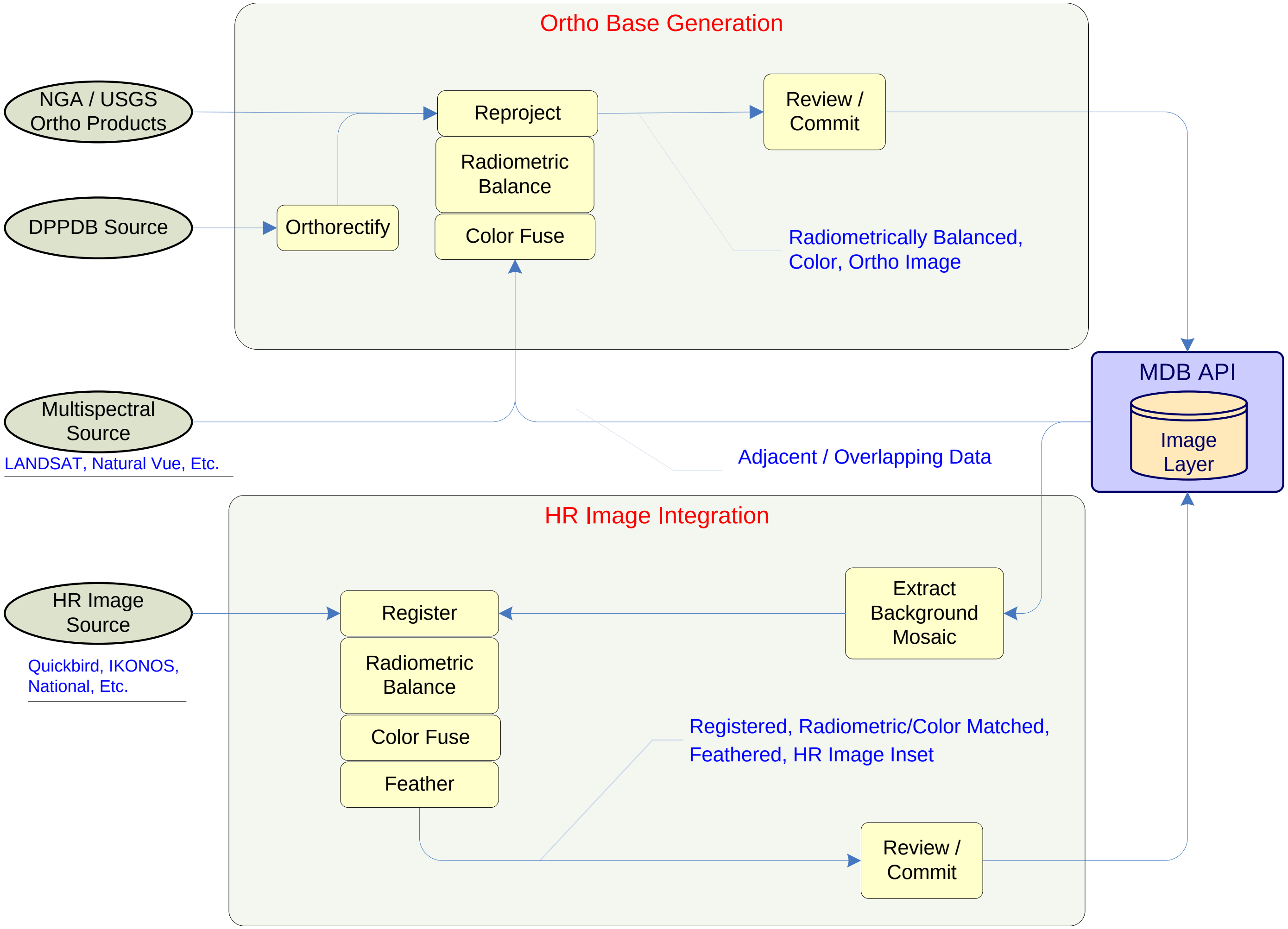
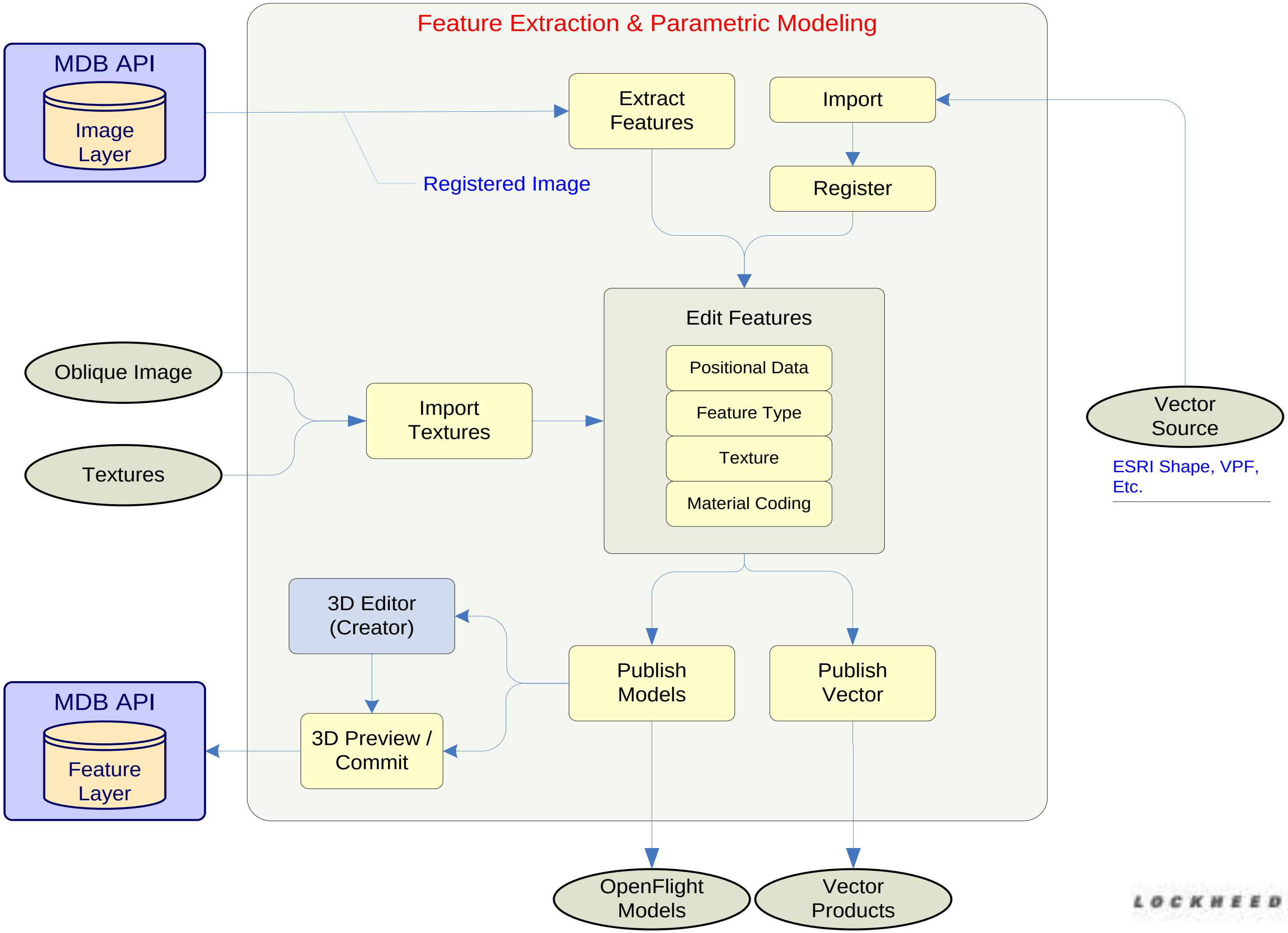


Image Integration



Ortho Base Generation
NGA / USGS Ortho Products
DPPDB Source
Orthorectify
Reproject
Radiometric Balance
Color Fuse
Review / Commit
Radiometrically Balanced, Color, Ortho Image
MDB API
Image Layer
Multispectral Source
LANDSAT, Natural Vue, Etc.
Adjacent / Overlapping Data
HR Image Integration
HR Image Source
Quickbird, IKONOS, National, Etc.
Register
Radiometric Balance
Color Fuse
Feather
Extract Background Mosaic
Registered, Radiometric/Color Matched, Feathered, HR Image Inset
Review / Commit

Feature Integration



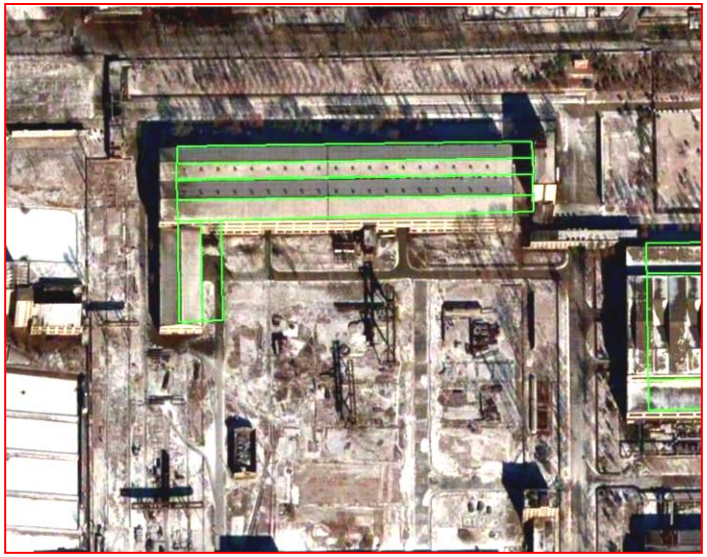
Vector Correction

Correction of Vectors to Imagery

- Ground Control Points (GCPs) used to correct vectors to MDB
- GCPs Automatically generated

Correction of Imagery to Vectors

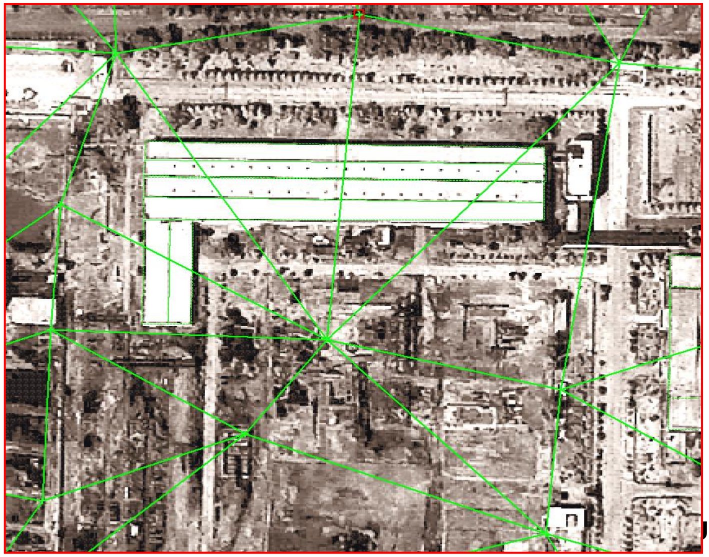
- MDB vector layers used as guide for new image registration
- New image insets do NOT invalidate existing vectors

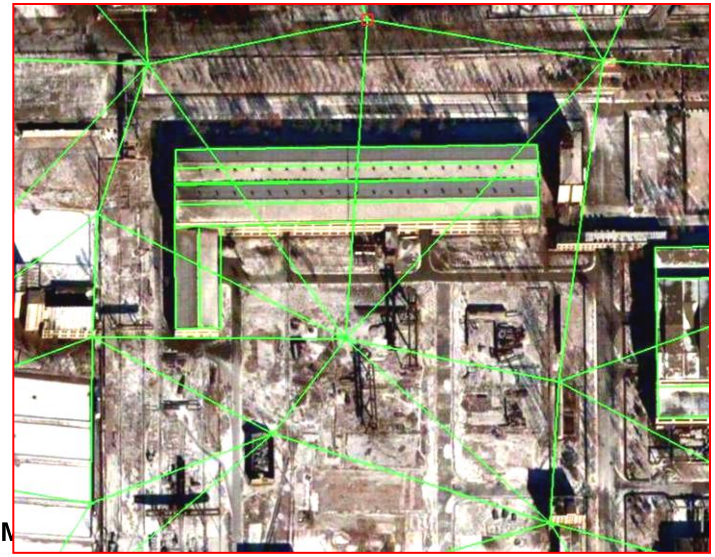


Top Right:
HR Image Inset

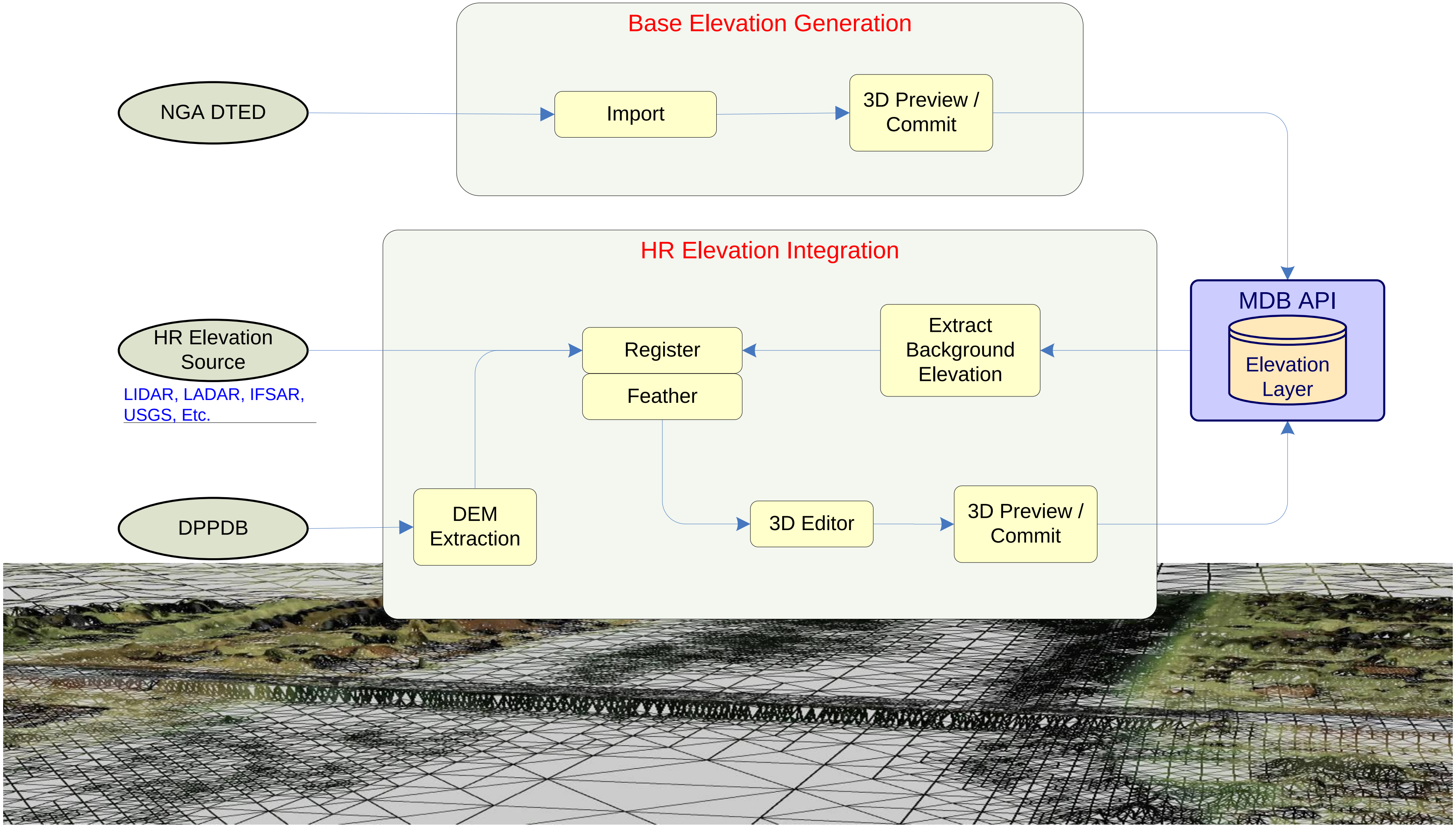
Bottom Left:
Mosaic Image

Bottom Right:
Registered HR Image Inset





Elevation Integration



Base Elevation Generation

NGA DTED

Import

3D Preview / Commit

HR Elevation Integration

HR Elevation Source

LIDAR, LADAR, IFSAR, USGS, Etc.

Register

Feather

Extract Background Elevation

MDB API

Elevation Layer

DPPDB

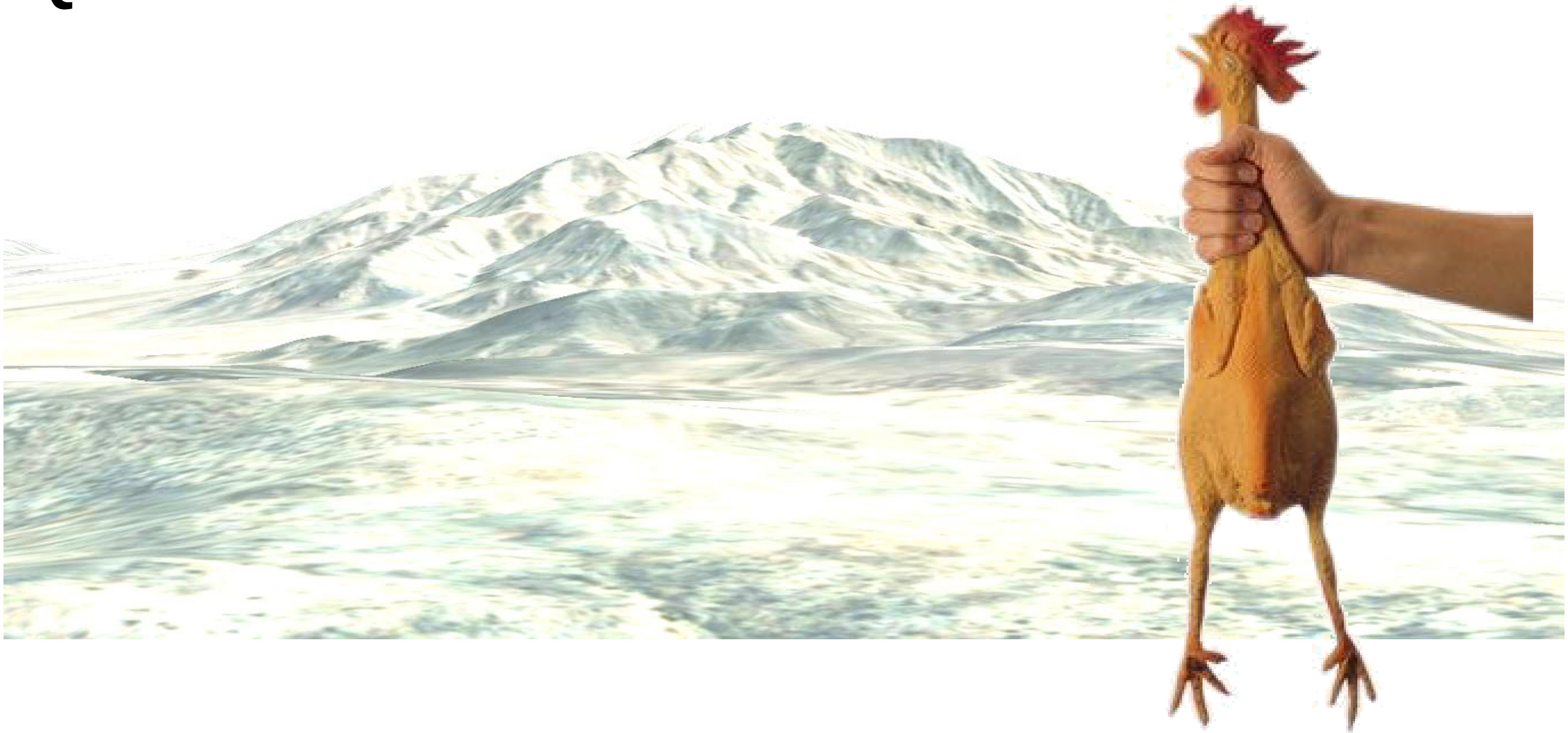
DEM Extraction

3D Editor

3D Preview / Commit

Get Off the Stage Slide

Questions?



Case Study #2 – U2MG Integration

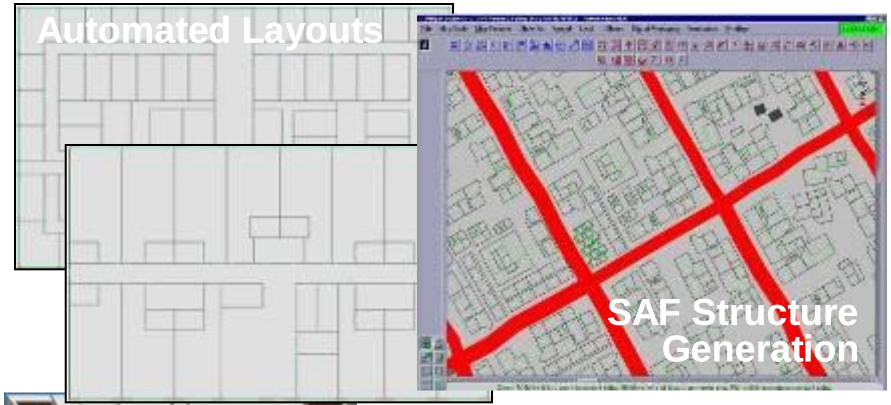
Applied Research Associates

U2MG Background

- Urban and Underground Model Generator (U2MG)
- Originally funded as BAA through RDECOM, Orlando
- Prototype based on Integrated Munitions Effects Assessment (IMEA) editor

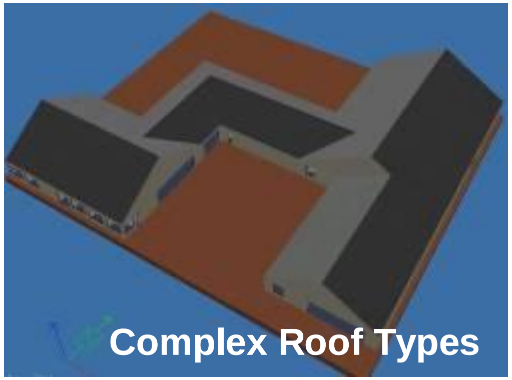
U2MG Feature Highlights

- Supports creation of geo-typical and geo-specific buildings
- Generates structurally correct buildings and places rooms, hallways, doors, windows, and furniture automatically
- Support for complex roof types such as Gable, Hip, Gambrel, and more
- Buildings are exported to visual (e.g. OpenFlight) and SAF formats (OneSAF Test Bed and OneSAF Objective System)


Automated Layouts

SAF Structure Generation

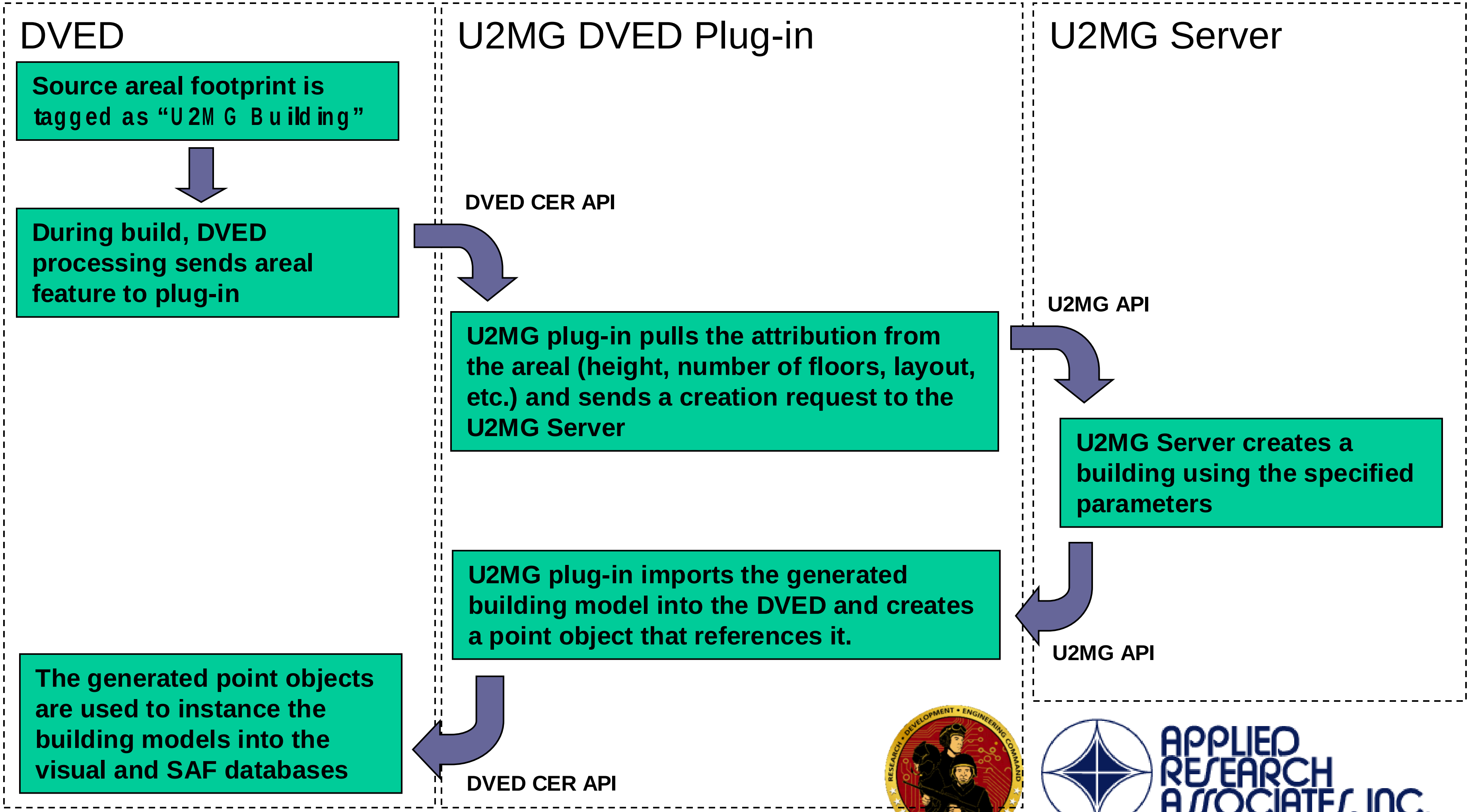
Automated Furniture Placement


Complex Roof Types

U2MG DVED Integration Outline

- U2MG has API for procedural generation of building models from areal and point features
  - C++ API encapsulates access to separate U2MG server process through a hybrid XML/COM interface
  - C++ API provides multiple parameters to govern building construction (number of floors, roof type, interior layout types, textures for visual representation, etc.)
- DVED U2MG plugin maps data from DVED to U2MG API parameters for each areal building outline and calls the U2MG API to construct the building
  - Visual OpenFlight models (with OTB MES attribution), OOS UHRB XML, and point features with orientation information to instance the models are returned to DVED for further processing

U2MG DVED Integration Flow



DVED

Source areal footprint is tagged as “U2MG Building”

During build, DVED processing sends areal feature to plug-in

DVED CER API

U2MG DVED Plug-in

U2MG plug-in pulls the attribution from the areal (height, number of floors, layout, etc.) and sends a creation request to the U2MG Server

U2MG API

U2MG Server

U2MG Server creates a building using the specified parameters

U2MG API

U2MG plug-in imports the generated building model into the DVED and creates a point object that references it.

DVED CER API

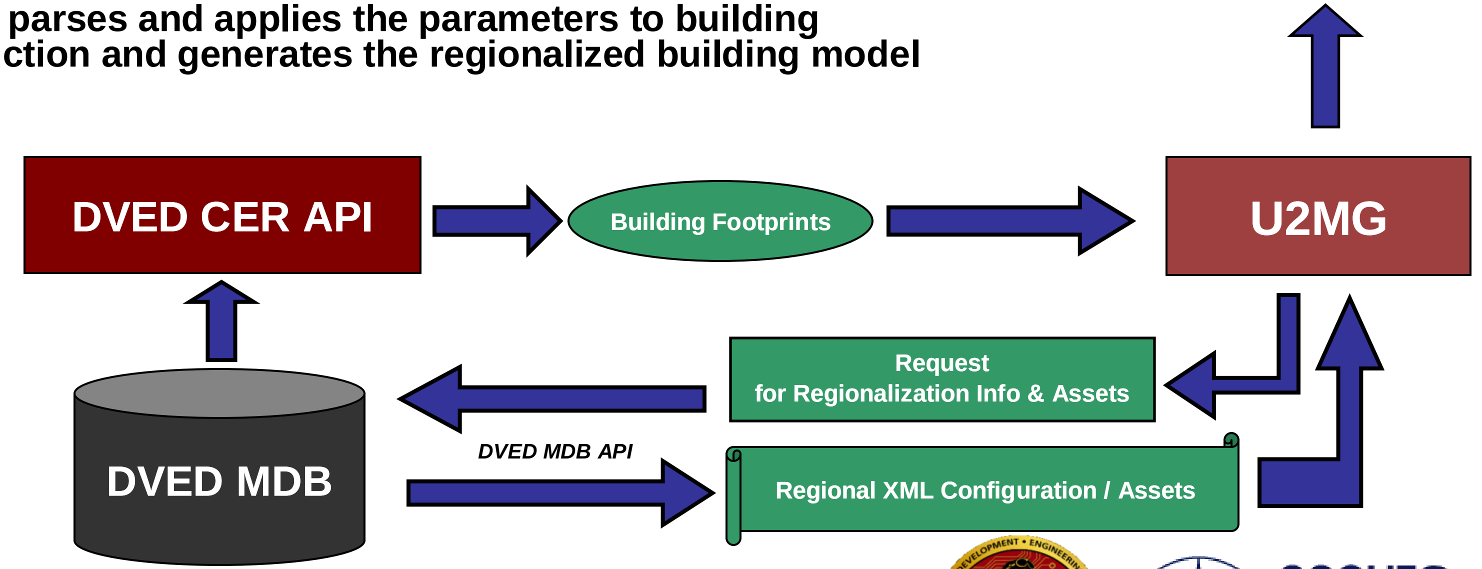
The generated point objects are used to instance the building models into the visual and SAF databases

U2MG DVED Regionalization

- **Regions of the world often have common construction materials for buildings and government specification on how buildings can be constructed**
  - **U2MG has parameters that govern**
    - Construction materials
    - Wall thicknesses
    - Visual Texturing
  - **DVED stores regionalization information for plugins that make use of it**
    - Regionalization is stored and layered by DVED, however each plugin can specify what information should be stored with regionalization information
- **The RUGUD program has designed and partially implemented a U2MG regionalization approach for integration into DVED**

U2MG DVED Regionalization Flow

- DVED Determines which region and priority is associated with each building areal, assigns regionalization GUID to attribution
- DVED Provides Building Footprints attributed with regionalization GUID
- U2MG queries MDB for regionalization XML associated with GUID
- U2MG parses and applies the parameters to building construction and generates the regionalized building model


DVED CER API
Building Footprints
U2MG
Request
for Regionalization Info & Assets
DVED MDB API
Regional XML Configuration / Assets
DVED MDB

U2MG DVED Summary

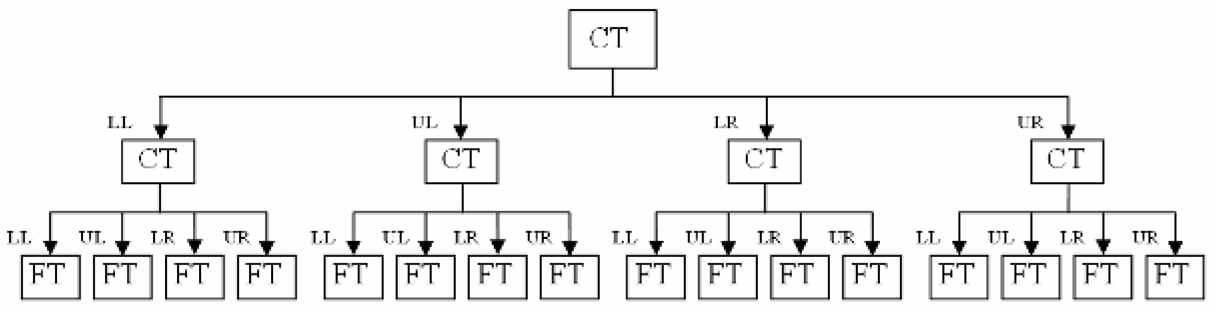
- **U2MG is fully integrated into DVED processing via CER API**
  - **Capable of generating thousands of unique buildings for each database**
- **Lessons Learned**
  - **CER API is .NET Capability**
    - Had to create COM solution to interoperate with .NET CER API
  - **CER API Documentation / Example Code Base**
    - Documentation missing / example code sparse
      - **Had to spend extra time searching through CER API headers for needed functionality**
  - **Multi-Machine Build Not Documented**
    - Had to spend large amount time with support to determine how to update plugin code to support MMB processing
  - **Process Management**
    - Weird cases causing plugin to crash takes down entire application
      - **COM solution isolates U2MG process and protects main application from crashing plugin**

Case Study #3 – Output Compiler Development

Output Compiler Development Overview

- **The SEE API provides:**
  - **An in memory representation of the 3D scene, along with texture coordinates**
  - **Material attributes for each polygon**
  - **Correlated vector layers for the 3D scene**
  - **Textures**
  - **Models**
  - **Regionalization rules**
  - **A complete API for dealing with geometry and spatial data**

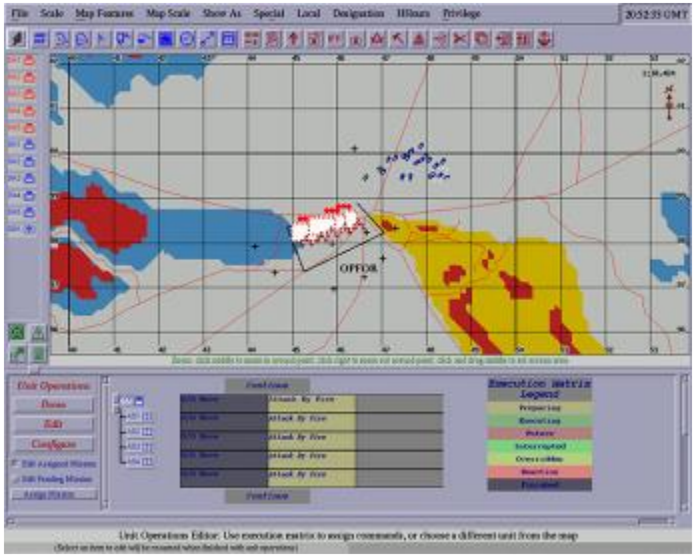
OTW Runtime Example



- To develop an Out the Window runtime output compiler, you have to map the in-memory scene graph to your own format.
  - Polygons
  - Materials
  - Texture Coordinates
- Models and Textures are available for you to copy or convert as needed
- This process is only as complex as the output format being written for

SAF Format Example

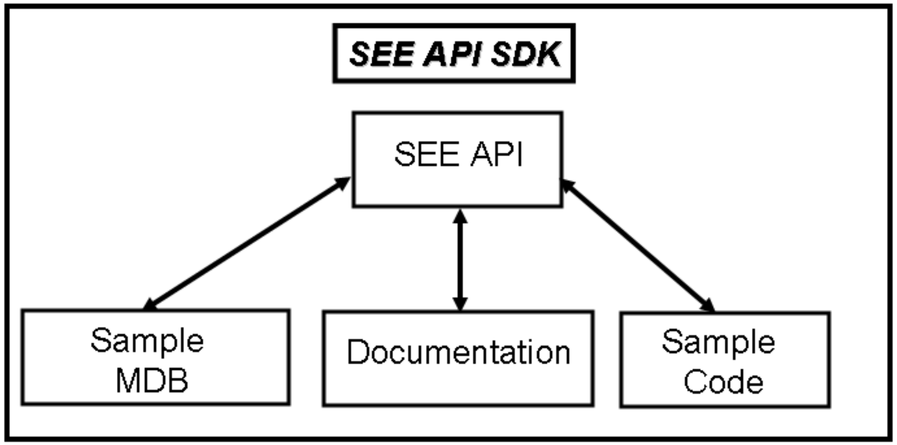
- In addition to the 3D Scene, attributed and correlated vector layers are provided
  - E.g. Road linears will be expanded to areals that match the polygons in the 3D scene
  - Just grab the layers needed and convert them to your format
- Because the database is built block by block, if the format being written doesn't support blocks of data, it will be necessary to stitch it together at the end of the build



Unit Operations Editor: Use execution matrix to assign commands, or choose a different unit from the map

SEE API SDK

- Sample MDB
  - Terra Vista project
  - Imagery
  - Elevation data
  - Vectors
  - Static and moving models with textures
  - Animation and Special Effect
- Documentation
  - Doxygen of the SEE API
  - User’s manual
  - UML diagrams of major modules
  - Data Representation Model
- Sample Code
  - Sample importer source code
  - Sample culture compiler source code
  - Sample output compiler source code (SEDRIS)



Developer Support Milestones

- The SDK and documentation will be ready for release in the end of March 2007
- The next developer’s forum will be in early summer 2007
- Email developersforum@secore-dved.com if you are interested in attending

Questions?